



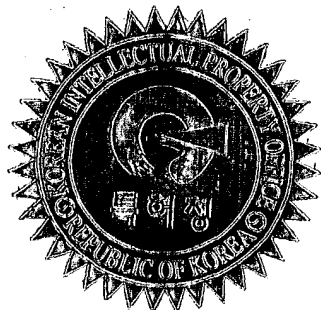
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 10-2002-0062913  
Application Number

출원년월일 : 2002년 10월 15일  
Date of Application OCT 15, 2002

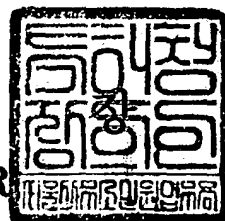
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003      07      07      일  
          년      월

특      허      청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2002.10.15
【발명의 명칭】	인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조와 디지털 콘텐츠 메타데이터 검 색 방법 및 장치
【발명의 영문명칭】	INDEX DATA STREAM STRUCTURE FOR METADATA OF DIGITAL CONTENTS INCLUDING ENCODED KEY INFORMATION, METHOD AND APPARATUS FOR METADATA OF DIGITAL CONTENTS USING THIS STRUCTURE
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	김동진
【대리인코드】	9-1999-000041-4
【포괄위임등록번호】	2002-007585-8
【발명자】	
【성명의 국문표기】	신효섭
【성명의 영문표기】	SHIN,Hyo Seop
【주민등록번호】	711013-1528617
【우편번호】	157-203
【주소】	서울특별시 강서구 가양3동 6단지 아파트 612동 1310호
【국적】	KR
【우선권주장】	
【출원국명】	KR
【출원종류】	특허
【출원번호】	10-2002-0043097
【출원일자】	2002.07.23
【증명서류】	미첨부
【심사청구】	청구

## 【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
김동진 (인)

## 【수수료】

【기본출원료】 20 면 29,000 원

【가산출원료】 21 면 21,000 원

【우선권주장료】 1 건 26,000 원

【심사청구료】 7 항 333,000 원

【합계】 409,000 원

## 【첨부서류】

1. 요약서·명세서(도면)\_1통 2.우선권증명서류 및 동 번역문[특허청 기재출]\_1통

**【요약서】****【요약】**

본 발명은 디지털 콘텐츠에 관한 메타데이터, 특히 TV-Anytime 포럼에서 정의하고 있는 디지털 콘텐츠 메타데이터(TVA 메타데이터)에 대한 인덱스 정보 스트림 구조와 디지털 콘텐츠 메타데이터 검색 방법 및 장치에 관한 것으로, 특히 인코딩된 키 정보를 제공함으로써 디지털 콘텐츠에 관한 메타데이터를 수신하는 수신장치가 메타데이터를 보다 신속하고 효율적으로 검색할 수 있도록 하는 인덱스 정보 스트림 구조와 디지털 콘텐츠 메타데이터 검색 방법 및 수신장치에 관한 것이다.

본 발명은 TVA 메타데이터가 프래그먼트 단위로 전송되는 전송 스트림 환경에서, TVA 메타데이터 프래그먼트에 대한 보다 효율적인 검색 및 접근 기능을 제공하기 위해 키 정보를 인코딩하는 접근 방법을 제시함으로써 TVA 메타데이터를 수신하는 수신장치가 TVA 메타데이터들에 대해 더욱 신속한 검색을 수행할 수 있도록 한다.

**【대표도】**

도 9

**【색인어】**

TV-Anytime, 메타데이터, 프래그먼트, 인코딩, 인덱스

**【명세서】****【발명의 명칭】**

인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조와 디지털 콘텐츠 메타데이터 검색 방법 및 장치{INDEX DATA STREAM STRUCTURE FOR METADATA OF DIGITAL CONTENTS INCLUDING ENCODED KEY INFORMATION, METHOD AND APPARATUS FOR METADATA OF DIGITAL CONTENTS USING THIS STRUCTURE}

**【도면의 간단한 설명】**

도 1은 일반적인 PDR의 개념도이다.

도 2는 일반적인 EPG 애플리케이션에서의 그리드 가이드 화면을 도시한다.

도 3은 TV-Anytime 포럼에서 정의하는 일반적인 메타데이터의 구조를 도시한다.

도 4는 TV-Anytime 포럼에서 정의하는 일반적인 프래그먼트의 개념도이다.

도 5는 TV-Anytime 포럼에서 정의하는 일반적인 컨테이너 개념도이다.

도 6은 종래의 인덱스 정보 스트림 구조를 도시한다.

도 7은 종래의 인덱스 정보의 자료 구조 및 검색과정을 도시한다.

도 8는 본 발명에 따른 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조를 도시한다.

도 9은 본 발명에 따른 인코딩된 키 정보를 이용한 인덱스 정보의 자료 구조 및 검색과정을 도시한다.

도 10은 본 발명에 따른 인코딩된 키 정보를 이용한 인덱스 정보 스트림에 대한 검색방법을 도시한다.

도 11는 본 발명에 따른 인덱스 정보 스트림을 수신하는 수신장치의 구조도이다.

도 12은 본 발명에 따른 인덱스 정보 스트림을 수신하는 수신장치의 동작 흐름도이다.

**【발명의 상세한 설명】**

**【발명의 목적】**

**【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<13> 본 발명은 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조와 디지털 콘텐츠 메타데이터 검색 방법, 그리고 상이 인덱스 정보 스트림을 수신하는 수신장치에 관한 것으로서, 특히 TV-Anytime 포럼에서 정의하고 있는 디지털 콘텐츠에 대한 XML 메타데이터(이하 'TVA 메타데이터')를 프래그먼트라는 독립적인 단위로 나누어 전송할 때, 키 정보에 대한 인코딩 구조를 제공함으로써 메타데이터를 수신하는 단말기가 인덱싱 정보를 활용한 프래그먼트에 대한 조건 검색을 보다 효율적으로 수행할 수 있도록 인덱스 정보 스트림 구조와 이를 이용한 TVA 메타데이터 검색방법 및 수신장치에 관한 것이다.

<14> TV-Anytime 포럼은, 개인용 대용량 저장장치를 가진 PDR(Personal Digital Recorder)과 같은 사용자 환경에서 오디오 비주얼 관련 서비스 제공을 위한 표준 개발을 목적으로 1999년 9월에 설립된 민간 표준 기구로서, 모든 사용자가 개인용 저장장치를 기반으로 자기가 원하는 방법으로 원하는 시간에 다양한 형태(기존의 방송 서비스 및 온라인 대화형 서비스 등)의 프로그램을 시청할 수 있게 하는 것을 그 구체적인 서비스 목표로 하고 있다.

- <15> TV-Anytime 포럼은 비즈니스 모델, 시스템/전송 인터페이스/컨텐츠 참조, 기술, 메타데이터, 컨텐츠 보호관리 등의 워킹 그룹으로 나누어 표준화를 진행하고 있으며, 본 발명과 관련된 메타데이터에 대해서는 2002년 6월 현재 "1st Draft of Metadata Specification SP003v1.3"까지 발표되었다.
- <16> 도 1을 참조하여 PDR의 구성을 간략하게 살펴보면, PDR(100)은 영상/음성 신호를 제공하는 제공자(200)로부터 공중파, 위성파, 인터넷망 등의 다양한 네트워크를 통해 영상/음성 신호 및 메타데이터를 수신하고, 필요에 따라 시청자의 시청 패턴, 기호 등을 수집하여 영상/음성 신호를 제공하는 제공자(200)에게 송신한다. PDR(100)은 수신된 영상/음성 신호 및 메타데이터를 저장하는 대용량 저장장치, 영상/음성 신호의 저장 및 재생을 위한 소프트웨어 및 영상/음성 신호에 대한 메타데이터를 검색/전시하기 위한 EPG(Electronic Program Guide) 애플리케이션을 포함하고, 사용자는 도 2에 도시된 EPG 애플리케이션의 그리드 가이드 화면 등을 통해 영상/음성 데이터에 관한 메타데이터 즉, 프로그램 제목, 프로그램 재생 시간 등을 확인하고 원하는 프로그램을 선택하여 네트워크를 통해 실시간으로 수신하거나 또는 이미 대량 저장장치에 저장된 영상/음성 데이터를 재생시킨다.
- <17> 메타데이터는 프로그램 타이틀과 개요와 같은 컨텐츠에 대한 설명적인 데이터를 의미하며, "데이터에 관한 데이터(data about data)"로 정의된다. TV-Anytime 포럼의 TVA 메타데이터 규격에서는 XML에 관한 표준기구 W3C의 표준인 XML Schema 언어(W3C의 XML 1.0 참조)를 사용하여 그 구조를 정의하는 한편, 각 메타데이터 엘리먼트 및 속성에 대한 의미(semantics)를 함께 규정하고 있다. 방송 컨텐츠에 관련된 TVA 메타데이터는 도 3에 도시된 바와 같이 "TVAMain" (300)이라는 루트 노드를 가지는 XML 문서로 구성되

며, 예를 들어 프로그램에 관한 메타데이터들은 “ProgramDescription” 노드 이하에서 프로그램 정보(ProgramInformation Table), 그룹 정보(GroupInformation Table), 프로그램 위치 정보(ProgramLocation Table), 서비스 정보(ServiceInformation Table) 등의 노드들로 구성된다.

<18> TV-Anytime 포럼은 방대한 TVA 메타데이터를 스트림으로 전송하기 위해 프래그먼트(fragment)라는 독립적인 단위로 TVA 메타데이터를 전송한다. 도 4를 참조하여 프래그먼트의 개념을 간략히 설명하면, 프래그먼트란 도 3에 도시된 XML 문서로 구성된 TVA 메타데이터를 소정의 트리 구조별로 분할한 것으로서, 예를 들면, 전체 TVA 메타데이터를 “TVAMain”을 상위 노드로 하여 소정의 자식 노드까지 포함하는 트리 구조 (프래그먼트 TVAMain), 프로그램 정보(ProgramInformation Table)를 상위 노드로 하여 이하 자식 노드까지 포함하는 트리 구조(프래그먼트 ProgramInformation), 브로드캐스트 이벤트 정보(BroadcastEvent)를 상위 노드로 하여 이하 자식 노드까지 포함하는 트리 구조(프래그먼트 BroadcastEvent)등으로 분할할 경우, 분할된 트리 구조 각각은 프래그먼트가 되고, 이 프래그먼트들은 다른 프래그먼트와 무관하게 독립적으로 전송가능하고 또한 개별 접근이 가능하다.

<19> 프래그먼트들에 대한 개별 접근을 위하여, 전송된 TVA 메타데이터 프래그먼트가 전체 메타데이터 트리 구조에서 어떤 노드를 참조하는지 즉, TVA 메타데이터



프래그먼트의 상위 노드가 어떤 노드에 해당하는지, 그리고 전송된 TVA 메타데이터 프래그먼트내에 포함된 키( '키' 는 프래그먼트가 참조하는 노드의 자식 노드를 의미하며, 사용자가 입력하게 되는 정보 필드(검색 조건), 예를 들면 '채널번호' 또는 '방송시간' 등이 키에 해당한다)들의 상기 TVA 메타데이터 프래그먼트내 상대 경로를 표시할 필요가 있으며, 이를 위해 W3C에서 정의한 XPath(XPath는 W3C에서 정의한 XML 문서내 노드에 대한 경로를 기술하는 신택스(syntax)임)를 사용한다.

<20> 한편, 프래그먼트들에 대한 효율적인 검색과 접근을 제공하기 위해서는 메타데이터 프래그먼트에 포함된 키에 대한 인덱스 구조가 별도로 필요하고, 이런 인덱스 구조에 관한 정보 즉, 인덱스 정보 또한 상기 메타데이터 프래그먼트들과는 독립적으로 전송된다.

<21> TV-Anytime 포럼이 제공하는 환경하에서, 소정 방송시간 조건에 해당하는 프로그램 정보를 사용자가 검색하고자 할 경우, 먼저 프래그먼트들과는 독립적으로 전송되는 인덱스 정보를 활용하여 원하는 방송시간 조건에 해당하는 메타데이터 프래그먼트의 위치(식별자)를 파악하고, 이 위치(식별자)를 기초로 해당 메타데이터 프래그먼트에 접근하여 상기 방송시간 조건에 해당하는 메타데이터를 추출하게 된다.

<22> 종래기술문헌(TV-Anytime Specification TV145, J.P. Evain, "1st Draft of Metadata Specification SP003v1.3," TV-Anytime 포럼 17차 회의, 몬트리올, 캐나다, 2002년 6월, 이하 '키 인덱스 기술문헌')에서는 메타데이터 프래그먼트 인덱스를 위한 인덱스 정보 스트림 구조를 제안하고 있다.

<23> 종래의 인덱스 정보 스트림 구조를 설명하기에 앞서 TV-Anytime 포럼이 정의하는 컨테이너 개념을 먼저 설명하기로 한다.

<24> TV-Anytime 포럼은 앞서 설명한 인덱스 정보와 메타데이터 프래그먼트 등의 모든 데이터가 전송되는 최상위 저장소 즉, 컨테이너(container)라는 최상위 전송 형태를 정의하는데, 컨테이너를 개략적으로 설명하면, 모든 컨테이너는 다수의 섹션들로 구성되고, 각 섹션에 상기 인덱스 정보와 메타데이터 프래그먼트들이 저장된다. 컨테이너는 운반하는 정보에 따라 키 인덱스 리스트(key\_index\_list) 섹션, 키 인덱스(key\_index) 섹션, 서브키 인덱스(sub\_key\_index) 섹션, 스트링 저장소(string\_repository) 섹션, 프래그먼트 데이터 저장소(fragment\_data\_repository) 섹션 등의 인덱스 정보 섹션을 운반하는 인덱스 컨테이너와 엘리먼트 테이블(elements\_table) 섹션, 스트링 저장소(string\_repository) 섹션, 프래그먼트 저장소(fragment\_data\_repository) 섹션 등의 메타데이터 프래그먼트 섹션을 운반하는 데이터 컨테이너로 구분된다. 위 구분은 컨테이너에 포함된 정보의 내용에 따라 구분하는 것으로서, 컨테이너의 구성은 인덱스 컨테이너와 데이터 컨테이너 모두 동일하다.

<25> 도 5을 참조하여 TV-Anytime 포럼에서 정의한 컨테이너를 살펴보면, 컨테이너는 미도시된 컨테이너 식별자 정보(container\_id) 필드와 다수의 섹션들을 포함하며, 각 섹션들은 'section\_id'의 인코딩 값에 따라 'section\_body'에 저장된 내용이 식별된다. 예를 들면, 'section\_id'의 인코딩 값이 '0X0004'인 섹션(10)은 키 인덱스 리스트(key\_index\_list) 섹션, 인코딩 값이 '0X0005'인 섹션(20)은 키 인덱스(key-index) 섹션, 인코딩 값이 '0X0006'인 섹션(30)은 서브키 인덱스(sub\_key\_index) 섹션, 인코딩 값이 '0X0001'인 섹션(40)은 엘리먼트 테이블(element\_table) 섹션, 인코딩 값이 '0X0003'인 섹션(50)은 프래그먼트 데이터 저장소(fragment\_data\_repository) 섹션으로 각각 식별된다.

- <26> TVA 메타데이터 프래그먼트는 데이터 컨테이너의 프래그먼트 데이터 저장소 (fragment\_data\_repository) 섹션(50)내에 저장되어 전송되는데, 데이터 컨테이너는 상기 데이터 컨테이너 내의 TVA 메타데이터 프래그먼트들에 대한 식별자 정보 (handle\_value) 필드를 엘리먼트 테이블 섹션(40)에 포함한다.
- <27> 결론적으로, TVA 메타데이터 프래그먼트는 TVA 메타데이터 프래그먼트가 포함된 컨테이너의 컨테이너 식별자 정보(container\_id)와 메타데이터 프래그먼트 식별자 정보 (handle\_value)에 의해 유일하게 식별된다.
- <28> 종래의 키 인덱스 기술문헌에서는 전술한 데이터 컨테이너에 저장된 TVA 메타데이터 프래그먼트를 인덱스하기 위한 인덱스 정보 스트림 구조 즉, 키 인덱스 리스트 (key\_index\_list) 섹션(10), 키 인덱스(key\_index) 섹션(20), 및 서브키 인덱스 (sub\_key\_index) 섹션(30)의 구조를 제안하고 있다. 상기 구조의 신택스(syntax)에 대해서는 키 인덱스 기술문헌에 자세히 기술되어 있으므로 이에 대한 구체적인 설명은 생략하기로 하고, 이하에서는 상기 구조를 인덱스 정보 스트림상의 세그먼트로 표현한 도 6을 참조하여 상기 구조를 살펴보기로 한다.
- <29> 종래의 인덱스 정보 스트림 구조에서 정의되는 키 인덱스 리스트(key\_index\_list) 섹션(10)은, 전송되는 모든 키의 리스트를 제공한다. 이 리스트에는 각 키들에 대한 키 정보(즉, 키가 포함된 메타데이터 프래그먼트에 대한 XPath(fragment\_xpath\_ptr)와, 키로서 사용되는 노드의 상기 프래그먼트의 XPath 위치에서의 관련 경로 즉, 키로 사용되는 노드들의 해당 프래그먼트내 상대 경로에 대한 XPath(key\_xpath\_ptr))와, 후술되는 키 인덱스(key\_index) 섹션(20)에 대한 식별 정보가 포함된다.

- <30>      상기 메타데이터 프래그먼트의 XPath는 TVA 메타데이터 XML 문서의 루트 노드에 대한 경로 즉, 절대 경로이며, 키로 사용되는 노드들의 XPath 즉, 키의 XPath는 상기 메타데이터 프래그먼트에 대한 키의 상대 경로를 나타낸다. 메타데이터 프래그먼트에 대한 XPath와 키에 대한 XPath는 각각 'fragment\_xpath\_ptr' 세그먼트(11)와 'key\_xpath\_ptr' 세그먼트(12)에 저장된다.
- <31>      또한, 키 인덱스 리스트(key\_index\_list) 섹션(10)에는 후술되는 각 키의 키 인덱스(key\_index) 섹션(20)에 대한 식별 정보(즉, 키 인덱스(key\_index) 섹션(20)이 저장된 컨테이너의 컨테이너 식별자 정보(container\_id) 및 키 인덱스 식별자 정보)가 포함되는데, 상기 컨테이너 식별자 정보 및 키 인덱스 식별자 정보들은 각각 키 인덱스 리스트(key\_index\_list) 섹션(10)의 'index\_container' 세그먼트 및 'key\_index\_identifier' 세그먼트에 저장되어 전송된다.
- <32>      종래의 인덱스 정보 스트림 구조에서 정의되는 키 인덱스(key\_index) 섹션(20)은, 후술되는 모든 서브키 인덱스(sub\_key\_index) 섹션(30)의 리스트를 제공하는데, 이 리스트에는 각 서브키 인덱스(sub\_key\_index) 섹션(30)들에 포함된 키값들의 범위를 나타내는 정보 즉, 각 서브키 인덱스(sub\_key\_index) 섹션(30)내 키값들 중 가장 큰 키값(이하 '대표키 값'), 및 각 대표키 값에 관련된 서브키 인덱스(sub\_key\_index) 섹션(30)에 대한 식별 정보(즉, 서브키 인덱스(sub\_key\_index) 섹션이 저장된 컨테이너의 컨테이너 식별자 정보(container\_id) 및 서브키 인덱스 식별자 정보)가 포함된다.
- <33>      키 인덱스(key\_index) 섹션(20)은 키 인덱스 리스트(key\_index\_list) 섹션(10)에서 정의된 키 인덱스 식별자 정보가 저장되는 'key\_index\_identifier' 세그먼트, 각 서브키 인덱스(sub\_key\_index) 섹션(30)의 대표키 값이 저장된 'high\_key\_value' 세그먼트

트(13), 대표키 값이 나타내는 범위에 해당하는 키값들을 포함하는 서브키 인덱스 (sub\_key\_index) 섹션(30)에 대한 식별 정보로서, 서브키 인덱스(sub\_key\_index) 섹션 (30)이 저장된 컨테이너의 컨테이너 식별자 정보(container\_id)와 서브키 인덱스 식별자 정보가 각각 저장된 'sub\_index\_container' 세그먼트 및 'sub\_index\_identifier' 세그먼트들을 포함한다.

<34> 종래의 인덱스 정보 스트림 구조에서 정의되는 서브키 인덱스(sub\_key\_index) 섹션(30)은, 해당 서브키 인덱스(sub\_key\_index) 섹션(30)에 포함된 키값들에 대한 리스트를 제공하는데, 이 리스트에는 해당 서브키 인덱스(sub\_key\_index) 섹션(30)에 포함된 키값들 및 상기 키값들을 가지는 메타데이터 프래그먼트에 대한 식별 정보(즉, 메타데이터 프래그먼트가 저장된 컨테이너의 컨테이너 식별자 정보 (container\_id) 와 상기 메타데이터 프래그먼트의 식별자 정보(handle\_value))를 제공한다.

<35> 서브키 인덱스(sub\_key\_index) 섹션(30)은 키 인덱스(key\_index) 섹션(20)에서 정의된 서브키 인덱스 식별자 정보가 저장되는 'sub\_index\_identifier' 세그먼트, 키값들이 저장되는 'key\_value' 세그먼트(14), 상기 키값을 가지는 메타데이터 프래그먼트에 대한 식별 정보로서 메타데이터 프래그먼트가 저장된 컨테이너의 컨테이너 식별자 정보(container\_id)와 프래그먼트 데이터 식별자 정보(handle\_value)가 각각 저장된 'target\_container' 세그먼트 및 'target\_handle' 세그먼트들을 포함한다.

<36> 상기 인덱스 정보 스트림 구조는 인덱스 정보를 예시적으로 도시한 도 7을 통해 더욱 쉽게 이해될 수 있다.

<37> 도 7은 채널 번호, 방송 시간 및 방송 재생 시간에 대한 키를 포함하는 키 인덱스 리스트(key\_index\_list) 섹션을 도시하고 있으며, 상기 채널 번호, 방송 시간 및 방송

재생 시간에 관한 키가 포함된 메타데이터 프래그먼트의 상위 노드는 도 3에 음영으로 도시된 'BroadcastEvent' (310)이다. 따라서, 'fragment\_xpath\_ptr' 세그먼트(11)에는 'BroadcastEvent' 프래그먼트에 대한 XPath ' /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent' 가 저장되고, 'key\_xpath\_ptr' 세그먼트(12)에는 'BroadcastEvent' 프래그먼트에 대한 채널 번호, 방송 시간, 및 방송 재생 시간의 키의 XPath인 '@ServiceId' (도 3의 311a), 'EventDescription/PublishedTime' (도 3의 311b) 및 'EventDescription/PublishedDuration' (도 3의 311c)이 각각 저장된다.

<38> 참고로, 도 7은 키 인덱스 리스트(key\_index\_list) 섹션 중 채널 번호(키의 XPath : @ServiceId)에 대한 키 인덱스(key\_index) 섹션(20) 및 서브키 인덱스(sub\_key\_index) 섹션(30)을 도시한다.

<39> 이러한 인덱스 정보 스트림 구조는, 메타데이터 검색을 위해 키가 선택되고 이에 따른 키 정보가 지정되어, 스트림으로 전송된 키 인덱스 리스트 내의 키 정보와 비교시, 양쪽의 XPath를 비교해야 하는 오버헤드를 발생시킨다. 이런 문제는 키 정보 중 프래그먼트로부터의 상대경로를 나타내는 키에 대한 비교에서도 마찬가지이지만, 특히 키보다 더욱 복잡한 프래그먼트의 비교에 있어 더욱 심각하다. 키 정보 중 프래그먼트에 대한 정보를 나타내는 프래그먼트 XPath는 그 기술 형태가 XML 문서 상에서 루트로부터 해당 노드까지의 경로(path)를 기술하는 방식이어서, 전송 비용이 비효율적일 뿐만 아니라, 단말기에서 XPath를 해석하는 비용도 크다. 예를 들어, TV-Anytime 프래그먼트 중에서 프로그램 위치 정보를 나타내는 브로드캐스트 이벤트 프래그먼트를 XPath로 표현하면 ' /TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent' 로 나타내어 진다

그런데, XPath는 XML 문서 상의 한 노드를 나타낼 때, 여러가지 방법으로 표현이 가능하다. 브로드캐스트 이벤트의 경우, 전술한 정규표현식 이외에도, /TVMain//BroadcastEvent, //BroadcastEvent 등으로 다양하게 표현되어 질 수 있다. 여기서 '/'는 XML문서 구조상 자식 노드임을 의미한다. 따라서 XPath 를 가지고 동일한 프래그먼트인지 검사하는 연산은 단순한 스트링들을 서로 매칭시키는 단순한 연산이 아니며, 특히 XPath경로가 축약되어 표시될 경우에는 해당 경로를 분석/비교해야 하는 오버헤드가 발생된다.

#### 【발명이 이루고자 하는 기술적 과제】

<40> 상술한 문제점을 해결하기 위해 제안된 본 발명은, 디지털 콘텐츠 메타데이터를 빠르게 인덱싱할 수 있도록 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조를 제공하는데 그 목적이 있다.

<41> 또한 본 발명은, 디지털 콘텐츠 메타데이터를 신속하게 인덱싱하도록 인코딩된 키 정보를 포함하는 인덱스 정보 스트림에 대한 검색방법 및 상기 인덱스 정보 스트림을 수신하는 수신장치를 제공하는데 그 목적이 있다.

#### 【발명의 구성 및 작용】

<42> 상술한 목적을 달성하기 위한 본 발명에 따른 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조는, 서브키 인덱스 섹션 식별 정보, 키 값들 및 상기 키 값들에 해당하는 메타데이터 프래그먼트의 식별 정보를 포함하는 서브키 인덱스(sub\_key\_index) 섹션; 키 인덱스 섹션 식별 정보, 상기 서브키 인덱스(sub\_key\_index) 섹션들에 포함된 키 값들의 범위를 나타내는 대표키 값 및 상기 대표키

값이 나타내는 범위에 해당하는 키값을 포함하는 서브키 인덱스 섹션 식별 정보를 포함하는 키 인덱스(key\_index) 섹션; 및 인코딩된 키 정보 및 키 인덱스 섹션 식별 정보를 포함하는 키 인덱스 리스트(key\_index\_list) 섹션을 포함한다.

<43> 또한, 상술한 목적을 달성하기 위한 본 발명에 따른 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조를 이용한 메타데이터 검색방법은, 검색에 필요한 키 정보 및 키값을 지정하는 단계; 상기 키 정보를 인코딩하는 단계; 상기 키 인덱스 리스트(key\_index\_list) 섹션에서, 상기 인코딩된 키 정보와 일치하는 키 인덱스(key\_index) 섹션들의 식별 정보를 추출하는 단계; 상기 추출된 키 인덱스(key\_index) 섹션들의 식별 정보를 가지는 키 인덱스(key\_index) 섹션에서, 상기 지정된 키값이 포함되는 대표키 값에 관련된 서브키 인덱스(sub\_key\_index) 섹션들의 식별 정보를 추출하는 단계; 및 상기 추출된 서브키 인덱스(sub\_key\_index) 섹션들의 식별 정보를 가지는 서브키 인덱스(sub\_key\_index) 섹션에서, 상기 지정된 키값에 해당하는 메타데이터 프래그먼트의 식별 정보를 추출하는 단계를 포함한다.

<44> 또한, 상술한 목적을 달성하기 위한 본 발명에 따른 인덱스 정보 스트림을 수신하는 수신장치는, 사용자의 검색 요청을 입력받고 검색 결과를 전시하는 EPG 애플리케이션; 및 상기 검색 요청에 따라 인코딩된 키 정보 및 키 값을 지정하고, 상기 인코딩된 키 정보 및 키 값에 해당하는 메타데이터 프래그먼트를 추출하여 상기 EPG 애플리케이션으로 전달하는 메타데이터 엔진을 포함한다.

<45> 따라서, TVA 메타데이터를 수신하는 수신장치가 메타데이터 프래그먼트들에 대한 검색을 인코딩된 키 정보를 이용하여 보다 효율적으로 수행할 수 있게 된다.



- <46> 이와 같은 특징을 갖는 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조 및 상기 인덱스 정보 스트림 구조를 이용한 검색방법, 그리고 상기 인덱스 정보 스트림 구조를 가지는 인덱스 정보 스트림을 수신하는 수신장치를 첨부된 도면을 참조하여 설명하면 다음과 같다.
- <47> 먼저, 전송한 데이터 컨테이너에 저장되어 오는 TVA 메타데이터 프래그먼트를 인덱스하기 위해, 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조, 즉 키 인덱스 리스트(key\_index\_list) 섹션(110), 키 인덱스(key\_index) 섹션(120), 및 서브키 인덱스(sub\_key\_index) 섹션(130)의 구조를 정의하는 신택스를 설명하고, 이후 상기 신택스에 의해 정의되는 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조에 대해 설명하기로 한다.
- <48> 본 발명에 따른 인덱스 정보 스트림 구조를 정의하는 신택스는, 종래의 키 인덱스 기술문헌에 정의된 신택스와 달리 fragment\_descriptor(), key\_descriptor() 와 같은 키 정보의 인코딩 개념을 위해 새롭게 도입된 구조를 포함하여, 키 인덱스 리스트(key\_index\_list) 섹션, 키 인덱스(key\_index) 섹션, 및 서브키 인덱스(sub\_key\_index) 섹션의 구조를 재편성한다.
- <49> 1. 키 인덱스리스트(key\_index\_list) 섹션
- <50> 키 인덱스 리스트(key\_index\_list) 섹션은 전송되는 모든 키의 리스트를 제공한다.
- <51> 종래 기술에서 키 인덱스 리스트(key\_index\_list) 섹션 안에서 프래그먼트 XPath를 나타내는 'fragment\_xpath\_ptr'는 fragment\_descriptor() 로 대체된다.

## &lt;52&gt; 【표 1】

신택스	비트수(변경가능)
key_index_list() {	
for (j=0; j<key_index_count; j++) {	
fragment_descriptor()	16
key_descriptor()	16
Index_container	16
key_index_identifier	8
}	
}	

- <53>     key\_index\_count: 전송되는 모든 키의 수 즉, 전체 XML 문서에 대한 인덱스 수를 상술한다.
- <54>     fragment\_descriptor(): 인덱싱될 목표 프래그먼트의 XPath 위치 또는 프래그먼트 타입과 같은 프래그먼트 그룹 정보를 기술한다.
- <55>     key\_descriptor(): 인덱싱될 목표 프래그먼트 그룹의 XPath 위치내에서 키의 XPath 또는 키의 타입과 같은 키 그룹 정보를 기술한다.
- <56>     index\_container: 지정된 키 인덱스(key\_index) 섹션이 존재하는 컨테이너를 식별한다.
- <57>     key\_index\_identifier: index\_container에 의해 상술된 컨테이너내에서 키 인덱스(key\_index) 섹션을 식별한다. index\_container와 key\_index\_identifier의 조합에 의해 키 인덱스(key\_index) 섹션이 유일하게 식별될 수 있다.
- <58>     2 . 프래그먼트 디스크립터(fragment descriptor)
- <59>     fragment\_descriptor는 빈번하게 사용되는 표준 프래그먼트 타입에 대하여 소정 비트(8 비트, 16 비트 등 임의의 비트로 인코딩 가능), 소정 코드로 인코딩하는 구조를 제공함과 동시에, 사용자가 정의하는 메타데이터 프래그먼트 타입에 대해서는 추가정보로

서 XPath를 기술할 수 있는 구조를 제공한다. 즉, fragment\_descriptor가 '0xFF' 인 경우에는 사용자 정의 프래그먼트를 나타내므로, 바로 다음에 해당 사용자 정의 프래그먼트에 대한 XPath가 기술된다.

<60> 【표 2】

신택스	비트수(변경가능)
fragment_descriptor() {	
fragment_type	8
if (fragment_type == 0xFF) {	
fragment_xpath_ptr	16
}	
}	

<61> fragment\_type : 인덱싱되는 프래그먼트의 타입을 나타낸다. 빈번하게 사용되는 표준 프래그먼트 타입들에 대해 인코딩 값을 부여한다. 만약 fragment\_type 이 0xFF의 인코딩 값을 가질 경우에는 추가정보로서 fragment\_xpath\_ptr이 부가된다.

<62> 표 3은 TV-Anytime에서 검색시 빈번하게 사용되는 프래그먼트(이하 '표준 프래그먼트') 타입에 대한 예시적인 인코딩 값을 나타내고 있으나, 본 발명에 있어서 표준 프래그먼트 타입 및 인코딩 값이 표 3에 한정되는 것은 아니며, 응용예에 따라서 더 확장될 수 있다.

## &lt;63&gt; 【표 3】

프래그먼트 타입 인코딩

인코딩값	내용
0x00	미지정
0x01	프로그램 정보(ProgramInformation) 프래그먼트
0x02	그룹 정보(GroupInformation) 프래그먼트
0x03	크레딧 정보(CreditsInformation) 프래그먼트
0x04	프로그램 리뷰(ProgramReview) 프래그먼트
0x05	세그먼트 정보(SegmentInformation) 프래그먼트
0x06	서비스 정보(ServiceInformation) 프래그먼트
0x07	브로드캐스트이벤트(BroadcastEvent) 프래그먼트
0xFF	사용자 지정
0x08-0x0E 0x10-0xFE	예비용

## &lt;64&gt; 3 . 키 디스크립터(key descriptor)

<65> key\_descriptor는 빈번하게 사용되는 키(이하 ‘표준 키’) 타입에 대하여 소정 비트로 인코딩하는 구조를 제공함과 동시에, 사용자가 정의하는 키 타입에 대하여 XPath로도 기술할 수 있는 구조를 제공한다. 예를 들어, key\_descriptor가 ‘0xFF’ 인 경우에는 사용자 정의 키를 나타내므로, 사용자가 정의하는 키에 대해서는 XPath가 추가정보로서 기술된다.

## &lt;66&gt; 【표 4】

신택스	비트수(변경가능)
key_descriptor() {	
key_type	8
if (key_type == 0xFF) {	
key_xpath_ptr	16
}	
}	

<67> key\_type : 인덱싱되는 키의 타입을 나타낸다. 빈번하게 사용되는 표준 키 타입들에 대해 인코딩 값을 부여한다. 만약 key\_type 이 ‘0xFF’ 의 인코딩 값을 가질 경우에는 추가정보로서 key\_xpath\_ptr이 부가된다.

<68> key\_xpath\_ptr: 키로서 사용되는 노드의 프래그먼트 XPath에 관련한 상대경로이다.

<69> 본 명세서에서는 표준 키에 대한 인코딩 값을 명시하지 않았으나, 표준 키에 대한 인코딩 값은 표 3의 프래그먼트 타입 인코딩과 유사한 구조를 가지는 것으로 이해될 수 있다.

<70> 4 . 키 인덱스(key\_index) 섹션

<71> 키 인덱스(key\_index) 섹션 및 서브 키 인덱스(sub\_key\_index) 섹션의 정의는 종래의 키 인덱스 기술문헌과 동일하므로 자세한 설명은 이하 생략한다.

<72> 【표 5】

신택스	비트수(변경가능)
key_index () {	
key_index_identifier	8
for (j=0; j< sub_index_count; j++) {	
high_key_value	16
sub_index_container	16
sub_index_identifier	8
}	
}	

<73> 5 . 서브 키 인덱스(sub\_key\_index) 섹션

<74> 【표 6】

신택스	비트수(변경가능)
sub_key_index () {	
sub_index_identifier	8
for (j=0; j< reference_count; j++) {	
key_value	16
target_container	16
target _ handle	16
}	
}	

- <75> 이하, 전술한 신텍스에 의해 정의되는 인덱스 정보 스트림 구조를 인덱스 정보 스트림상의 세그먼트로 표현한 도 8을 참조하여 상기 구조를 살펴보기로 한다.
- <76> 인덱스 정보 스트림 구조에서 정의되는 키 인덱스 리스트(key\_index\_list) 섹션(110)은, 전송되는 모든 키의 리스트를 제공한다. 건 이 리스트에는 각 키들에 대한 키 정보(즉, 키가 포함된 메타데이터 프래그먼트에 대한 XPath를 인코딩한 인코딩 값(fragment\_descriptor)과, 키로서 사용되는 노드들의 상기 메타데이터 프래그먼트의 XPath 위치에서의 관련 경로, 즉 키 XPath를 인코딩한 인코딩 값(key\_descriptor))와, 후술되는 키 인덱스(key\_index) 섹션(120)에 대한 식별 정보가 포함된다. 참고로, 종래의 인덱스 정보 스트림 구조와 마찬가지로, 상기 메타데이터 프래그먼트의 XPath는 TVA 메타데이터 XML 문서의 루트 노드에 대한 경로 즉, 절대 경로이며, 키로 사용되는 노드의 XPath 즉, 키 XPath는 상기 메타데이터 프래그먼트에 대한 키의 상대 경로를 나타낸다. 본 발명에서 메타데이터 프래그먼트에 대한 XPath를 인코딩한 인코딩 값과 키 XPath를 인코딩한 인코딩 값은 각각 'fragment\_descriptor' 세그먼트(111)와 'key\_descriptor' 세그먼트(112)에 저장된다.
- <77> 전술한 바와 같이, 본 발명에서는 키 정보 중 메타데이터 프래그먼트에 대한 XPath에 대해 빈번하게 사용하는 표준 프래그먼트 타입일 경우, 종래의 프래그먼트의 XPath(fragment\_xpath\_ptr)를 소정의 코드로 인코딩한 인코딩 값(fragment\_descriptor)으로 제공한다. 빈번하게 사용되는 표준 프래그먼트 타입으로는, 예를 들어, 프로그램 정보(ProgramInformation), 프로그램 그룹 정보(GroupInformation), 크레딧 정보(CreditsInformation), 프로그램 리뷰 (ProgramReview), 세그먼트 정보(SegmentInformation), 브로드캐스트이벤트(BroadcastEvent), 서비스 정보

(ServiceInformation) 등이 있다. 이러한 프래그먼트 타입들에 대한 메타데이터 프래그먼트 XPath를 간단히 인코딩된 값으로 나타낼 수 있다면 메타데이터 검색에 따른 오버헤드를 감소시킬 수 있다.

<78> 따라서, 본 발명에 따른 인덱스 정보 스트림 구조에서는 표준 메타데이터 프래그먼트의 XPath를 인코딩 값으로 인코딩하여 저장한다. 또한, 사용자가 추가적으로 메타데이터 프래그먼트에 관한 키 정보를 XPath로서 정의할 수 있도록 모든 인코딩 값을 프래그먼트들에 할당하지 않고 일부 인코딩 값(예를 들면, '0xFF')을 사용자가 정의하는 메타데이터 프래그먼트에 관한 키 정보로 할당한다. 이 경우 예를 들면, 메타데이터 프래그먼트에 대한 XPath를 지정할 수 있는 추가 영역(표 2의 'fragment\_xpath\_ptr')이 제공된다.

<79> 표 3과 같이 인코딩된 실시예에서, 키 정보 중 메타데이터 프래그먼트에 대한 정보는, '0x01', '0x02', '0x03' 등의 인코딩된 값이며, '0x01'로 인코딩된 메타데이터 프래그먼트에 대한 정보는 '프로그램 정보(ProgramInformation) 프래그먼트'의 XPath를 나타낸다. 또한 메타데이터 프래그먼트에 대한 키 정보가 '0xFF'인 경우에는 사용자가 정의하는 메타데이터 프래그먼트를 의미하므로, 추가적으로 메타데이터 프래그먼트의 XPath를 지정할 수 있는 추가 영역이 제공된다.

<80> 위 실시예는 메타데이터 프래그먼트에 대해서만 서술하였으나, 상기 메타데이터 프래그먼트에 대한 키에 대해서도 동일하다. 빈번하게 사용되는 키에 대해서는 종래의 키 XPath가 아닌 인코딩된 값을 지정하여 사용할 수 있으며, 또한 인코딩된 값 중 소정의 값인 경우에는, 사용자가 추가적으로 키에 관한 XPath를 지정할

수 있다. 전술한 메타데이터 프래그먼트 XPath의 인코딩과 키 XPath의 인코딩은 동시에 사용되거나 또는 서로 독립적으로 사용되어 질 수 있다.

<81> 또한, 키 인덱스 리스트(key\_index\_list) 섹션(110)에는 후술되는 각 키의 키 인덱스(key\_index) 섹션(120)에 대한 식별 정보(즉, 키 인덱스(key\_index) 섹션(120)이 저장된 컨테이너의 컨테이너 식별자 정보(container\_id) 및 키 인덱스 식별자 정보)가 포함되는데, 상기 컨테이너 식별자 정보 및 키 인덱스 식별자 정보들은 각각 키 인덱스 리스트(key\_index\_list) 섹션(110)의 'index\_container' 세그먼트 및 'key\_index\_identifier' 세그먼트에 저장되어 전송된다.

<82> 키 인덱스(key\_index) 섹션(120) 및 서브키 인덱스(sub\_key\_index) 섹션(130)은, 종래의 키 인덱스 문헌과 동일하므로 이에 대한 설명은 생략한다.

<83> 상기 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조는 상기 인덱스 정보를 예시적으로 도시한 도 9를 참조하여 자세히 설명된다.

<84> 도 9은 채널 번호에 대한 '브로드캐스트이벤트(BroadcastEvent)' 프래그먼트의 XPath가 '0x07'로 인코딩된 키 인덱스 리스트 섹션(110)을 도시하고 있으며, 키 인덱스(key\_index) 섹션(120) 및 서브 키 인덱스(sub\_key\_index) 섹션(130)은 도 7에서 설명한 바와 같다.

<85> 이러한 인덱스 정보 스트림 구조는 빈번하게 사용되는 프래그먼트, 예를 들어, 프로그램 정보(ProgramInformation), 그룹 정보(GroupInformation), 브로드캐스트이벤트(BroadcastEvent) 등의 프래그먼트에 관련된 키 들을 사용할 때 매우 효과적이며, 수신 장치에 있어서 전체적인 오버헤드를 감소시킬 수 있다.



- <86> 인코딩된 키 정보를 포함하는 인덱스 정보 스트림 구조에서, 검색 조건에 부합하는 메타데이터 프래그먼트를 얻기 위해 도 10에 도시된 검색 방법이 수행된다.
- <87> 먼저 검색에 필요한 키 정보 및 키 값을 지정한다. (S100). 만약 검색 조건에 해당하는 메타데이터 프래그먼트 또는 키가 인코딩될 수 있는 표준 프래그먼트 또는 표준 키일 경우 상기 키 정보 즉, 메타데이터 프래그먼트의 XPath 또는 키 XPath는 소정의 코드값으로 인코딩된다(도 9에서, '브로드캐스트이벤트(BroadcastEvent)' 프래그먼트는 '0X07' 로 인코딩됨). 만약 인코딩 값에 의해 식별될 수 없는 프래그먼트 또는 키인 경우에는 종래와 같이 XPath로 표현된 키 정보를 지정한다(S200)
- <88> 지정된 키 정보 및 키값에 해당하는 메타데이터 프래그먼트를 찾기 위해, 전송 스트림의 키 인덱스 리스트(key\_index\_list) 섹션(110)에서 지정된 키 정보와 일치하는 키 인덱스(key\_index) 섹션(120)의 식별 정보를 추출한다(S300).
- <89> 다음 단계로서, 추출된 키 인덱스(key\_index) 섹션의 식별 정보를 이용하여 키 인덱스(key\_index) 섹션(120)을 찾아, 상기 지정된 키 값이 포함되는 대표키 값에 관련된 서브키 인덱스(sub\_key\_index) 섹션(130)의 식별 정보를 추출한다(S400).
- <90> 다음 단계로서, 추출된 서브키 인덱스(sub\_key\_index) 섹션(130)의 식별 정보를 이용하여 서브키 인덱스(sub\_key\_index) 섹션(130)을 찾은 후, 상기 지정된 키값에 해당하는 메타데이터 프래그먼트의 식별 정보를 추출한다(S500).
- <91> 마지막으로, 추출된 메타데이터 프래그먼트의 식별 정보를 이용하여 해당 데이터 컨테이너에서 메타데이터 프래그먼트를 추출하고(S600), 추출된 메타데이터 프래그먼트에 따라 검색 결과를 출력한다.

- <92> 도 11는 본 발명에 따른 인덱스 정보 스트림을 수신하는 수신장치를 도시한다.
- <93> 상기 수신장치는, 사용자의 검색 요청을 입력받아 메타데이터 엔진(500)에 전달하고, 메타데이터 엔진(500)으로부터 검색 결과를 수신하여 전시하는 EPG 애플리케이션(400), 검색 요청에 따라 인코딩된 키 정보 및 키 값을 지정하고, 인코딩된 키 정보 및 키 값에 해당하는 메타데이터 프래그먼트를 전송 스트림(600)으로부터 추출하고, 추출된 메타데이터 프래그먼트를 처리하여 EPG 애플리케이션(400)에 출력하는 메타데이터 엔진(500)으로 구성된다.
- <94> 상기 메타데이터 엔진(500)은, 입력된 키 정보 및 키 값에 해당하는 메타데이터 프래그먼트의 식별 정보를 전송 스트림(600)으로부터 추출하는 인덱스 매니저(510), 입력된 메타데이터 프래그먼트의 식별 정보에 해당하는 메타데이터 프래그먼트를 전송 스트림(600)으로부터 추출하는 프래그먼트 매니저(520), 및 검색 요청에 따른 인코딩된 키 정보 및 키 값을 지정하고, 지정된 키 정보 및 키 값을 인덱스 매니저(510)로 전달하고, 인덱스 매니저(510)에서 추출된 메타데이터 프래그먼트의 식별 정보를 프래그먼트 매니저(520)로 전달하고, 그리고, 프래그먼트 매니저(520)에서 추출된 메타데이터 프래그먼트를 처리하여 검색 결과를 출력하는 질의 처리부(530)를 포함한다.
- <95> 도 12은 도 11의 수신장치에 의한 인덱스 정보 스트림에 대한 검색방법을 도시한다.
- <96> EPG 애플리케이션(400)을 통해 사용자가 메타데이터 검색을 요청하면(S1100), 질의 처리부가(530)가 검색에 필요한 키 정보 및 키 값을 지정한다(S1200).

- <97> 키 정보는 소정 코드에 따라 인코딩되어(S1300), 인덱스 매니저(510)로 입력되고, 인덱스 매니저(510)는 상기 인코딩된 키 정보 및 키 값을 이용하여 전송 스트림(600)에서 키 인덱스 리스트(key\_index\_list) 섹션, 키 인덱스(key\_index) 섹션 및 서브키 인덱스(sub\_key\_index) 섹션을 수신하고, 상기 지정된 키값에 해당하는 메타데이터 프래그먼트의 식별 정보를 추출한다(S1400).
- <98> 추출된 메타데이터 프래그먼트의 식별 정보는 질의처리부(530)을 통해, 또는 질의처리부(530)를 거치지 않고 직접 프래그먼트 매니저(520)로 전달된다(S1500).
- <99> 다음 단계로서, 프래그먼트 매니저(520)는 상기 추출된 메타데이터 프래그먼트의 식별 정보를 이용하여 전송 스트림(600)에서 메타데이터 프래그먼트를 추출하여 질의처리부(530)로 출력한다(S1600).
- <100> 마지막으로, 질의처리부(530)는 메타데이터 프래그먼트를 처리하여(S1600) 검색결과를 EPG 애플리케이션(400)으로 출력하고, EPG 애플리케이션(400)은 사용자에게 전시한다(S1700)
- <101> 본 발명은 도면에 도시된 일 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

#### 【발명의 효과】

- <102> 본 발명은 메타데이터가 프래그먼트 단위로 전송되는 전송 스트림 환경에서, 메타데이터 프래그먼트를 신속하게 검색하기 위해 메타데이터 프래그먼트들에 대해 단순화

된 인덱싱을 제공하는 인덱스 정보 스트림 구조, 인덱스 정보 스트림에 대한 검색 방법 및 인덱스 정보 스트림을 수신하는 수신장치를 제시하였다.

<103>      본 발명을 통하여, 메타데이터들에 대한 신속한 검색이 가능하여 수신장치의 오버헤드가 감소하여 검색 시간이 단축되고 수신장치의 효율이 증대된다.

**【특허청구범위】****【청구항 1】**

서브키 인덱스 섹션 식별 정보, 키값들, 및 상기 키값들에 해당하는 메타데이터 프래그먼트의 식별 정보를 포함하는 서브키 인덱스(sub\_key\_index) 섹션;

키 인덱스 섹션 식별 정보, 상기 서브키 인덱스(sub\_key\_index) 섹션들에 포함된 키값들의 범위를 나타내는 대표키 값, 및 상기 대표키 값이 나타내는 범위에 해당하는 키값을 포함하는 서브키 인덱스 섹션 식별 정보를 포함하는 키 인덱스(key\_index) 섹션; 및

인코딩된 키 정보 및 키 인덱스 섹션 식별 정보를 포함하는 키 인덱스 리스트(key\_index\_list) 섹션을 포함하는 것을 특징으로 하는 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조.

**【청구항 2】**

제 1항에 있어서, 상기 인코딩된 키 정보는, 표준 메타데이터 프래그먼트의 XPath를 인코딩한 인코딩 값인 것을 특징으로 하는 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조.

**【청구항 3】**

제 1항에 있어서, 상기 인코딩된 키 정보는, 표준 키의 XPath를 인코딩한 인코딩 값인 것을 특징으로 하는 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조.

**【청구항 4】**

제 1항에 있어서, 상기 인코딩된 키 정보가 특정 코드값일 경우, XPath로 표현된 별도의 키 정보를 더 포함하는 것을 특징으로 하는 인코딩된 키 정보를 포함하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조.

**【청구항 5】**

제 1항 내지 제 4항중 어느 한 항에 따른 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조를 이용한 메타데이터 검색방법에 있어서,

검색에 필요한 키 정보 및 키값을 지정하는 단계;

상기 키 정보를 인코딩하는 단계;

상기 키 인덱스 리스트(key\_index\_list) 섹션에서, 상기 인코딩된 키 정보와 일치하는 키 인덱스(key\_index) 섹션들의 식별 정보를 추출하는 단계;

상기 추출된 키 인덱스(key\_index) 섹션들의 식별 정보를 가지는 키 인덱스 (key\_index) 섹션에서, 상기 지정된 키값이 포함되는 대표키 값에 관련된 서브키 인덱스 (sub\_key\_index) 섹션들의 식별 정보를 추출하는 단계; 및

상기 추출된 서브키 인덱스(sub\_key\_index) 섹션들의 식별 정보를 가지는 서브키 인덱스(sub\_key\_index) 섹션에서, 상기 지정된 키값에 해당하는 메타데이터 프래그먼트의 식별 정보를 추출하는 단계를 포함하는 것을 특징으로 하는 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조를 이용한 메타데이터 검색방법

**【청구항 6】**

제 1항 내지 제 4항중 어느 한 항에 따른 디지털 콘텐츠 메타데이터의 인덱스 정보 스트림 구조를 가지는 인덱스 정보 스트림을 수신하는 수신장치에 있어서,

사용자의 검색 요청을 입력받고 검색 결과를 전시하는 EPG 애플리케이션; 및

상기 검색 요청에 따라 인코딩된 키 정보 및 키 값을 지정하고, 상기 인코딩된 키 정보 및 키 값에 해당하는 메타데이터 프래그먼트를 추출하여 상기 EPG 애플리케이션으로 전달하는 메타데이터 엔진을 포함하는 것을 특징으로 하는 인덱스 정보 스트림 수신 장치.

**【청구항 7】**

제 6항에 있어서, 상기 메타데이터 엔진은,

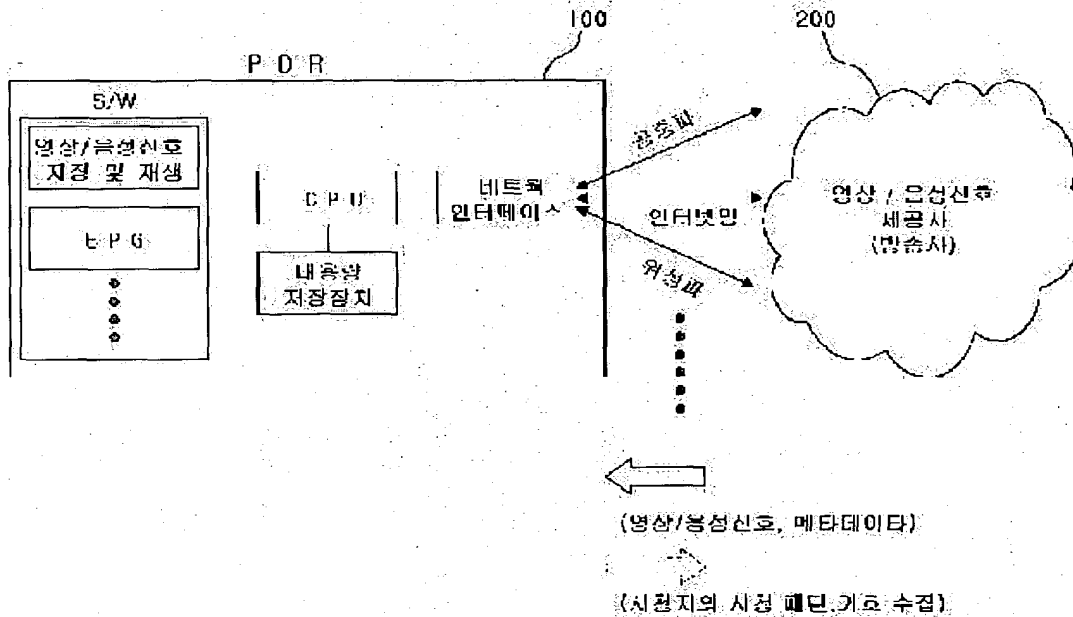
입력된 키 정보 및 키값에 해당하는 메타데이터 프래그먼트의 식별 정보를 추출하는 인덱스 매니저;

입력된 메타데이터 프래그먼트의 식별 정보에 해당하는 메타데이터 프래그먼트를 추출하는 프래그먼트 매니저; 및,

상기 인코딩된 키 정보 및 키값을 지정하고, 상기 인코딩된 키 정보 및 키값을 상기 인덱스 매니저로 전달하고, 상기 인덱스 매니저에서 추출된 메타데이터 프래그먼트의 식별 정보를 상기 프래그먼트 매니저로 전달하고, 그리고, 상기 프래그먼트 매니저에서 추출된 메타데이터 프래그먼트를 처리하여 상기 EPG 애플리케이션으로 검색 결과를 출력하는 질의 처리부를 포함하는 것을 특징으로 하는 인덱스 정보 스트림 수신장치.

【도면】

【도 1】

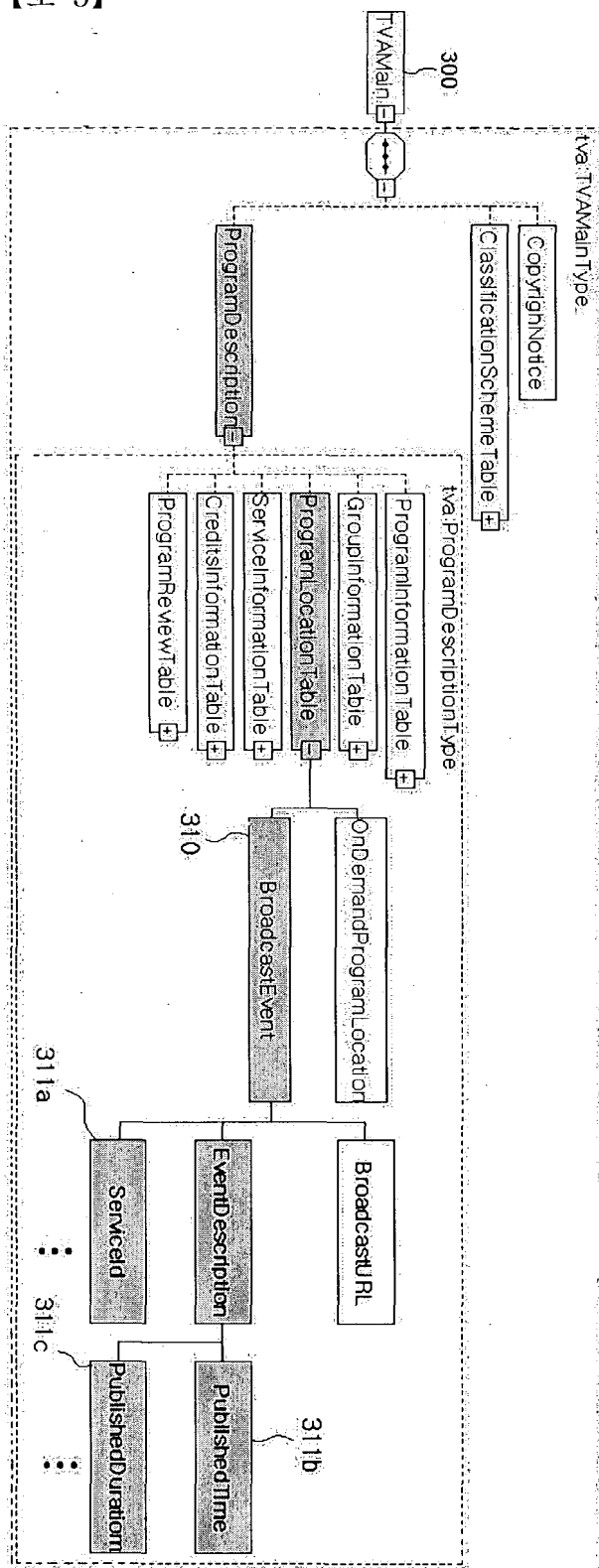


【도 2】

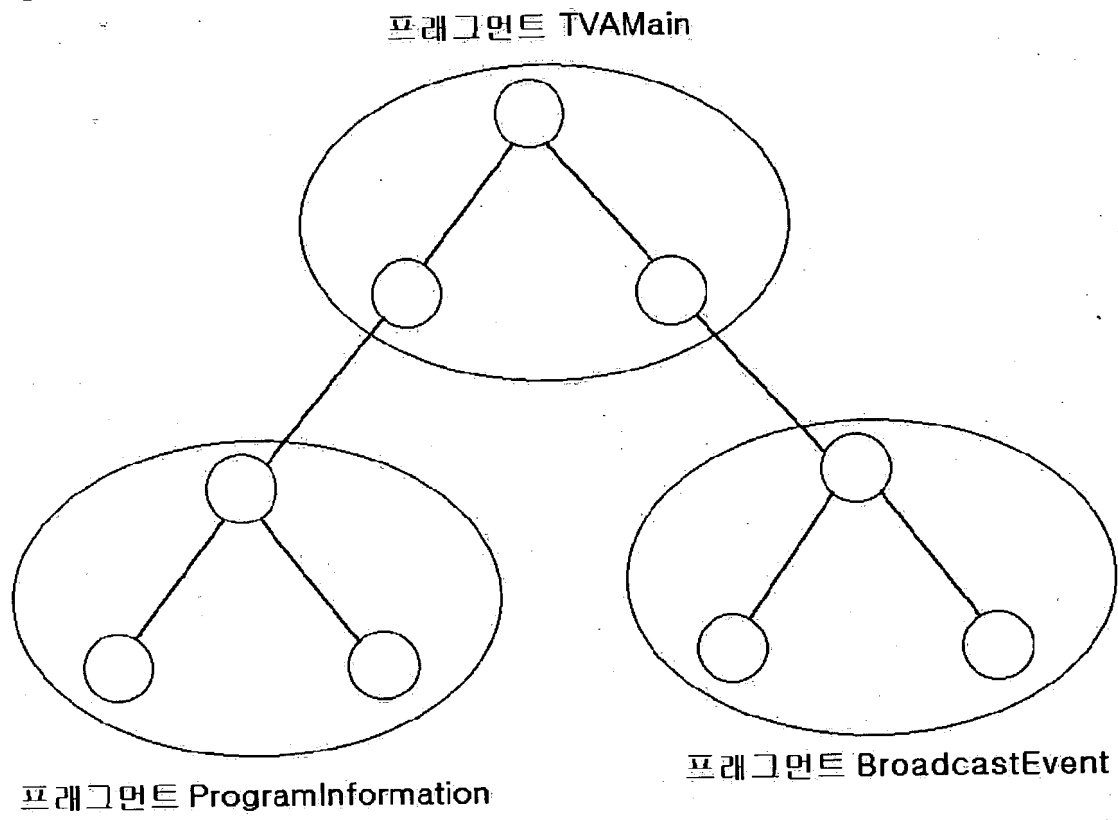
Today	9:00p	9:30p	10:00p
507 HBOF	◀ Out of Africa - Movie - Spanish <b>HD</b> ▶		
508 HBFW	◀ Into the West - Movie		Hocus Pocus ▶
512 MAX	◀ Vertical Limit - Movie <b>HD</b>		American Be... ▶
513 MMAX	The Gauntlet - Movie		
514 MaxW	◀ The Bridges of Madison County - Movie <b>LB</b>		



【도 3】

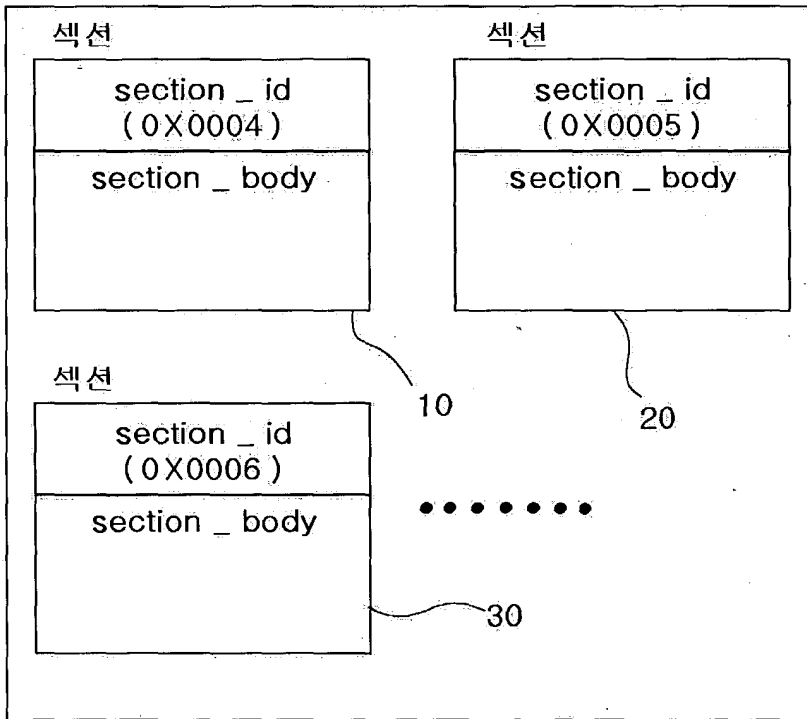


【도 4】

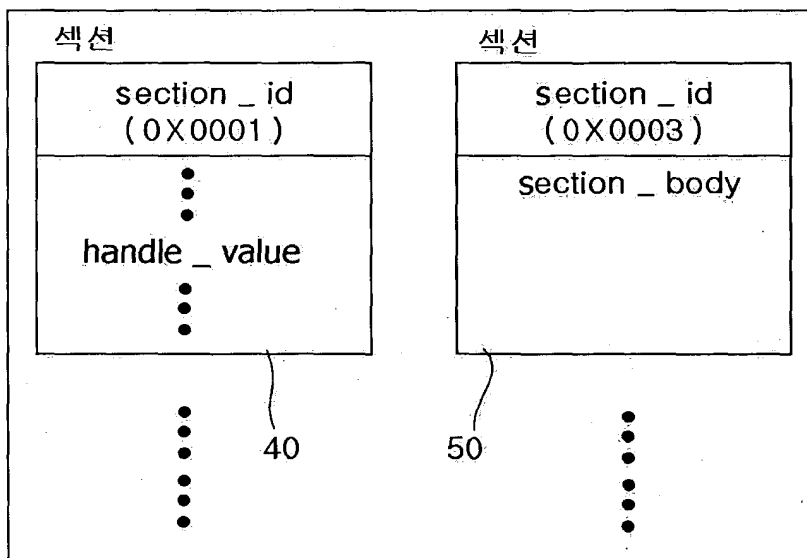


## 【도 5】

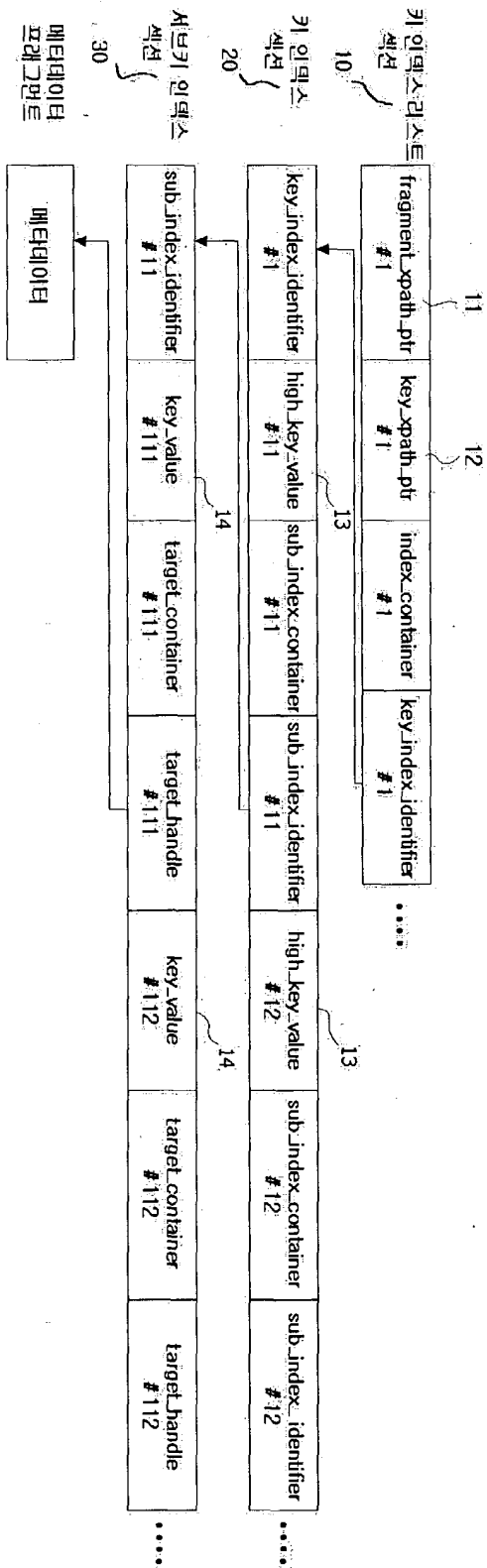
## 인덱스 컨테이너



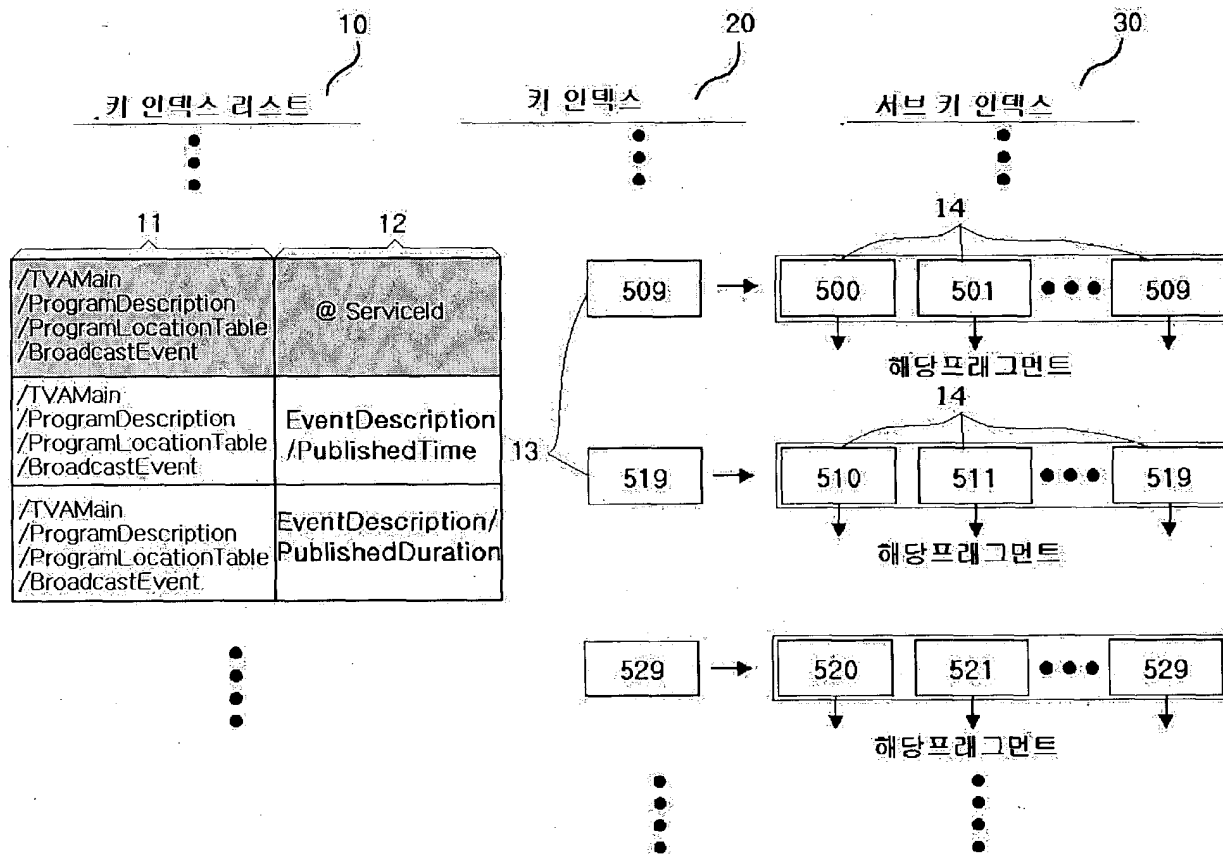
## 데이터 컨테이너



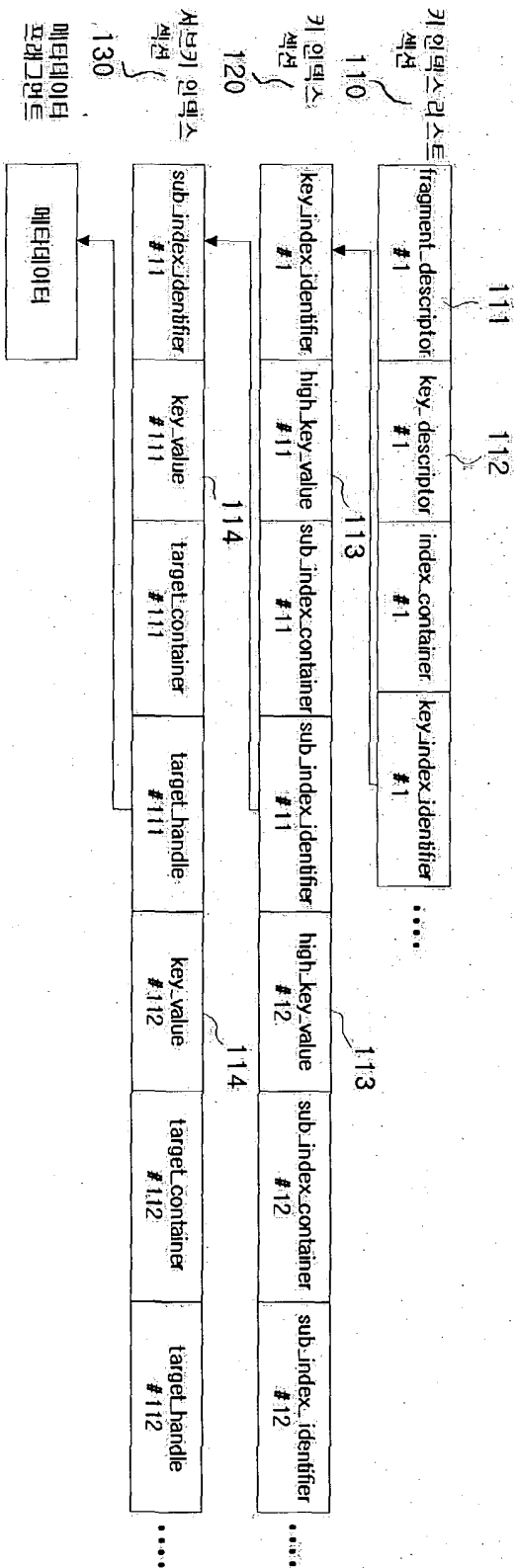
【부 6】



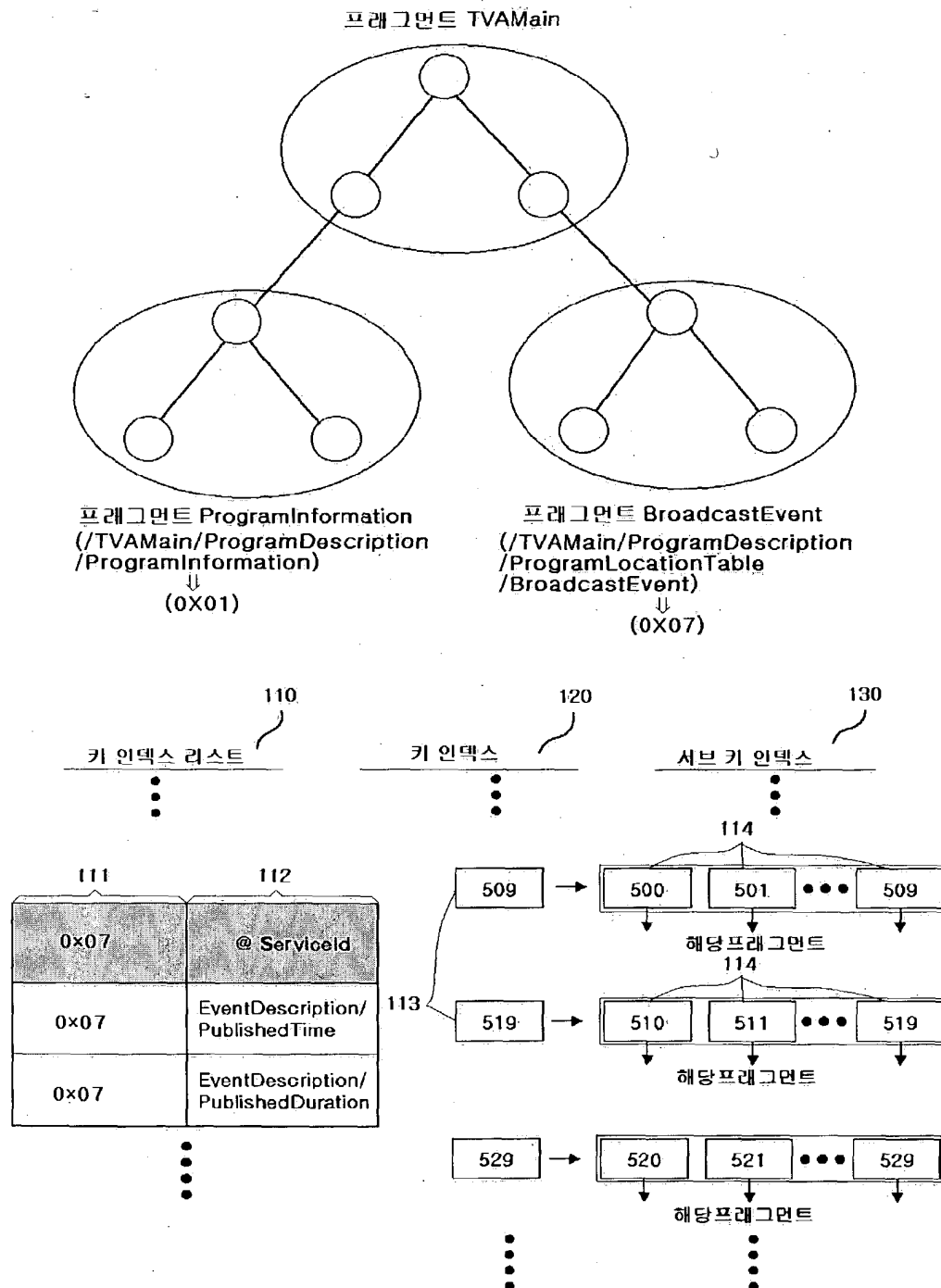
【도 7】



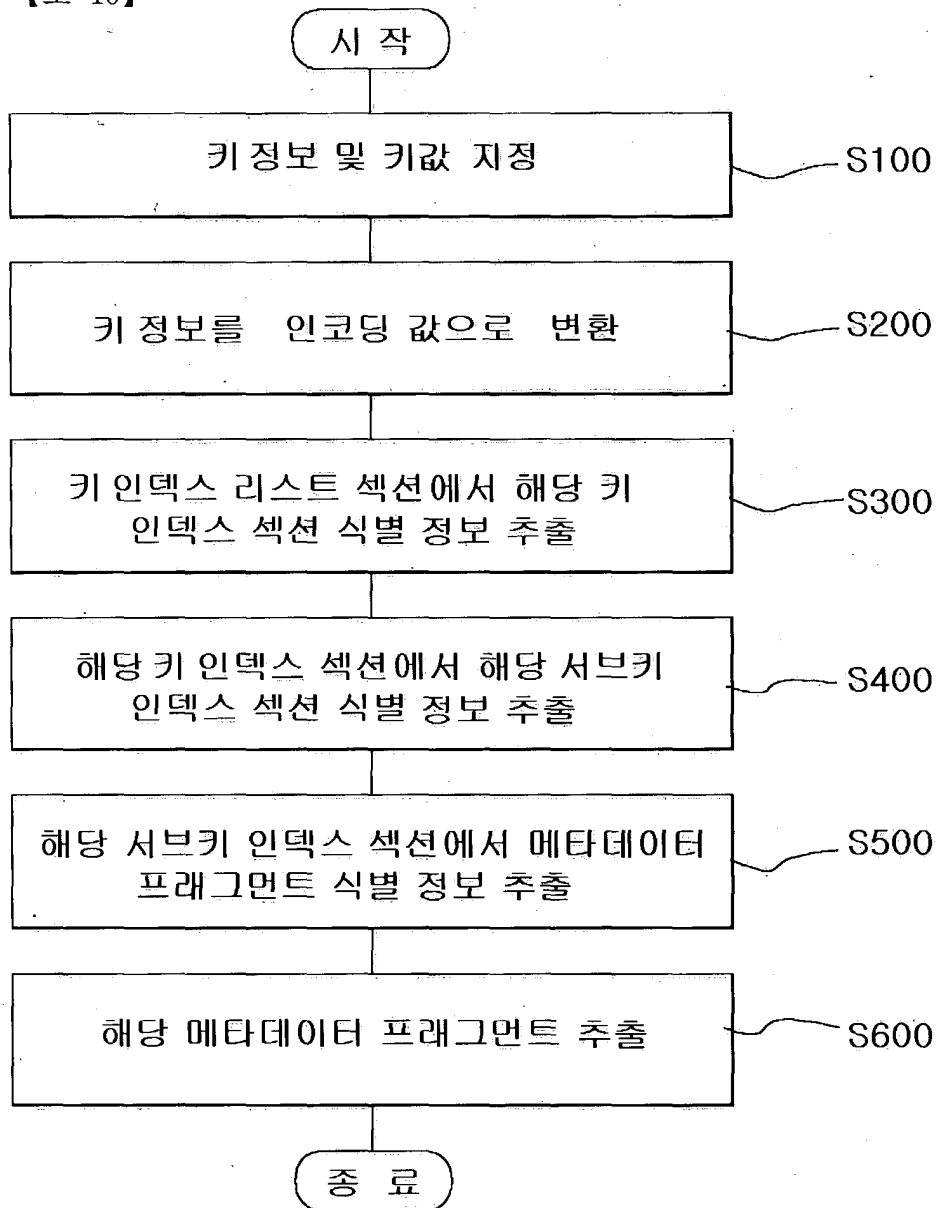
【 8 】



【도 9】

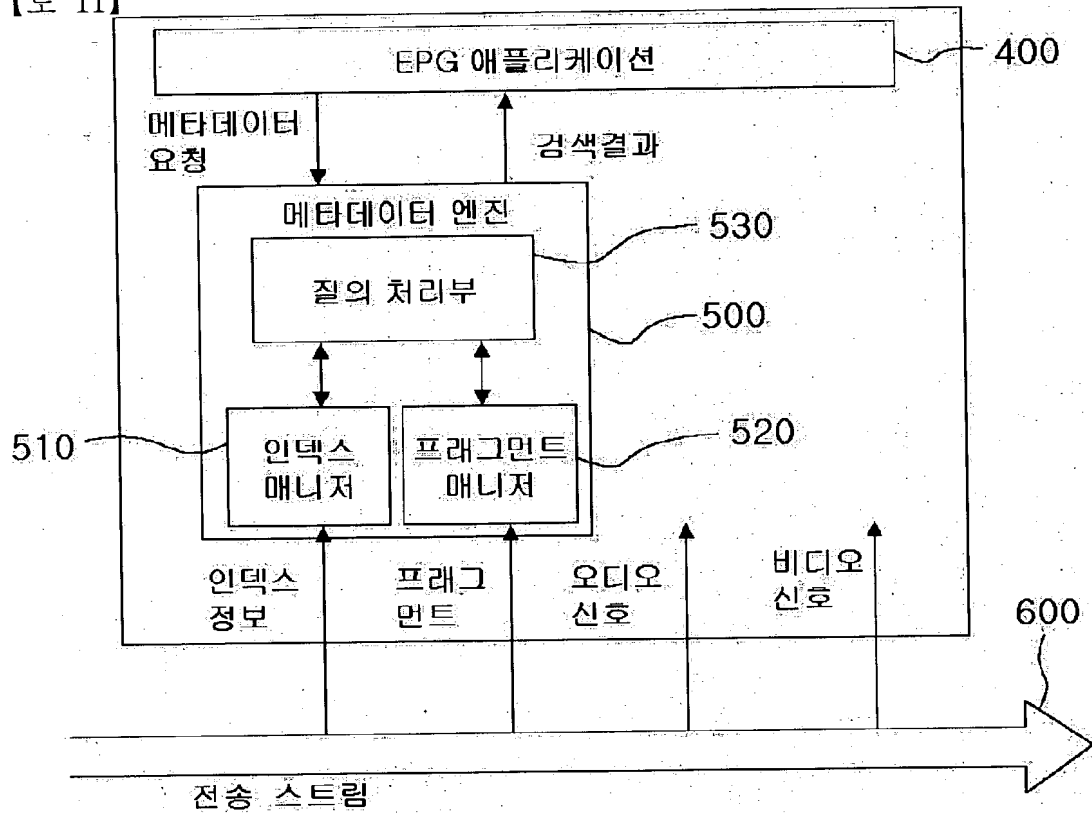


【도 10】





【도 11】



【도 12】

